AMENDMENTS TO CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for automatic service composition, searching services from registered service specifications to find a single service or compose a service flow according to a service request, the method comprising:

a service request receiving step, for receiving a problem file established according to the service request;

a service specification receiving step, for receiving a domain file established according to at least one service specification, the at least one service specification being used for executing an action which defines an action name, zero or at least one input parameter, and zero or at least one output parameter, wherein any two different service specifications can use an object with the same data type as the input parameter or the output parameter;

a new object predicting step, for predicting a new object by extracting data types needed by the declared objects of the problem file or the domain file to select at least one service specification related to the data type and storing the selected service specifications in a chosen service set;

a new object declaring step, for declaring the new object by counting a frequency N for each data type used as the input parameter and a frequency M for each data type used as the output parameter in the chosen service set; if M>0, the data type is also used as the output parameter,

and Cx(N+M) new objects are declared in the domain file for the data type, wherein C is an integer; ~~and~~

a service composition generating step, for generating a service flow by generating a series of action execution sequences of the single service or composite service, from service specifications stored in the service repository according to the problem file and the domain file, for being executed to accomplish the service request; and

a correlation establishing step, for establishing at least one level of data-type-service graph between all service specifications and data types, wherein the new object predicting step follows the interactive usage correlation structure to select at least one service specification related to the data type;

wherein the service request is a file with an XML (extensible markup language) format and is translated into a problem file with a PDDL (planning domain definition language) format via a translation process.

2. (Canceled)

3. (Original) The method claimed in claim 1, wherein the service request defines an initial state and a goal state and uses the series of action execution sequences to transform from the initial state to the goal state to accomplish the service request.

4. (Canceled)

5. (Original) The method claimed in claim 1, wherein the service specifications are files with an XML format and

are translated into a domain file with a PDDL format via a translation process.

6. (Previously presented) The method claimed in claim 5, wherein a DMAL-S standard is used to define the service specifications, which includes definitions of a plurality of preconditions and effects, as a file based on a RDF (resource description framework) format.

7. (Previously presented) The method claimed in claim 5, wherein a WSDL (web services description language) standard is used to define the service specifications, which includes definitions of a service address, an action name, a plurality of input parameters and a plurality of output parameters, as a file with an XML format.

8. (Original) The method claimed in claim 1, wherein the service specifications are defined by at least one service provider and are published in a service repository of a service registry.

9. (Original) The method claimed in claim 8, wherein a UDDI (universal description discovery and integration) registration protocol is used to publish the service specifications to the service registry.

10. (Original) The method claimed in claim 1, wherein the service specification defines zero or at least one precondition and zero or at least one effect.

11. (Original) The method claimed in claim 10, further comprising, after the new object declaring step, a modifying step, for adding a precondition and an effect to each output parameter of each service specification in the chosen service set.

12. (Currently Amended) A system for automatic service composition, searching services from registered service specifications to find a single service or compose a service flow according to a service request, the system comprising:

a service repository managed by a service registry, storing at least one service specification, each service specification being used for executing an action which defines an action name, zero or at least one input parameter, and zero or at least one output parameter, wherein any two different service specifications can use an object with the same data type as the input parameter or the output parameter;

a composition engine comprising:

a service request translation module for translating the service request to a problem file;

a service specification translation module for translating the service specification to a domain file and extracting data types needed by the declared objects of the problem file or the domain file to select at least one service specification related to the data type and storing the selected service specifications in a chosen service set, then counting a frequency N for each data type used as the input parameter and a frequency M for each data type used as the output parameter in the chosen service set; if M>0, the data type is also used as the output parameter,

and Cx(N+M) new objects are declared in the domain file for the data type, wherein C is an integer;

a planning module for generating a series of action execution sequences of the single service or composite service, from service specifications stored in the service repository according to the problem file and the domain file, for being executed to accomplish the service request; and

a service composition translation module for translating the composite service to a service flow document; and

[[a]] an execution engine for executing the service specification defined by the service flow document to accomplish the service request;

wherein the service specification translation module is also used for establishing at least one level of data-type-service graph between all service specifications and data types and following the interactive usage correlation structure to select at least one service specification related to the data type, and wherein the service request is a file with an XML format and translated into a problem file with a PDDL format via a translation process.

13. (Original) The system claimed in claim 12, wherein the composition engine is stored with one service registry.

14. (Original) The system claimed in claim 12, wherein the service specifications are defined by at least one service provider and published in a service repository of a service registry.

15 (Canceled)

16. (Original) The system claimed in claim 12, wherein the service specifications are files with an XML format and are translated into a domain file with a PDDL format via a translation process.

17. (Previously presented) The system claimed in claim 16, wherein a DMAL-S standard is used to define the service specifications, which includes definitions of a plurality of preconditions and effects, as a file based on a RDF format.

18. (Previously presented) The system claimed in claim 16, wherein a WSDL standard is used to define the service specifications, which includes definitions of a service address, an action name, a plurality of input parameters and a plurality of output parameters, as a file with an XML format.

19. (Original) The system claimed in claim 12, wherein the service specifications are defined by at least one service provider and are published in a service repository of a service registry.

20. (Original) The system claimed in claim 12, wherein the service specification defines zero or at least one precondition and zero or at least one effect.

21. (Original) The system claimed in claim 20, wherein the service specification translation module is also used for adding a precondition and an effect to each output

parameter of each service specification in the chosen service set.

22. (Canceled)

23. (Original) The system claimed in claim 12, wherein the service request defines an initial state and a goal state and uses the series of action execution sequences to transform from the initial state to the goal state to accomplish the service request.